What is claimed is:

1. A multilevel cache system, comprising:

a first data array;

a second data array coupled to the first data array; and

a merged tag array coupled to the second data array.

2. The multilevel cache system of claim 1, wherein the merged tag array is further coupled to a processor state control component.

3. The multilevel cache system of claim 1, wherein the merged tag array is configured to contain directory information for the first data array and second data array.

4. The multilevel cache system of claim 3, wherein:

the first data array contains a plurality of sets and the second data array contains a plurality of sets;

the merged tag array contains a plurality of entries, each of which corresponds to a set in the first data array and to one or more sets in the second data array; and

each entry in the merged tag array contains a presence field indicating whether the corresponding set in the second data array contains a copy of information present in a corresponding set in the first data array.

5. The multilevel cache system of claim 4, wherein:

the second data array further contains a plurality of ways;

the merged tag array further contains a presence-way field and a plurality of tag fields, each tag field corresponding to a second data array way; and

each presence-way field indicates which, if any, second data array way contains a copy of information present in a corresponding set in the first data array.

6. The multilevel cache system of claim 5, further comprising a single level translation lookaside buffer coupled to the merged tag array, wherein the single level translation lookaside buffer contains all available memory address translations.

7. A merged tag array having a plurality of entries, each of said entries comprising:

a first directory field containing information about the contents of a corresponding set in a first data array; and

a second directory field containing information about the contents of a corresponding set in a second data array.

8. The merged tag array of claim 7, wherein:

the first directory field comprises a presence field for indicating whether a corresponding set in the first data array contains the same information as a corresponding set in the second data array; and

the second directory field comprises a tag field corresponding to sets in the second data array.

9. The merged tag array of claim 8, wherein:

the second data array contains a plurality of ways;

each entry in the merged tag array contains a plurality of second directory fields, each of which correspond to a different way in the second data array; and

each entry in the merged tag array further contains a plurality of presence-way fields for indicating which way in the second data array contains a copy of information present in a corresponding set in the first data array.

10. A multilevel cache system, comprising:

a first array for storing data;

a second array for storing data; and

a third array for storing tags for both the first array and second array.

11. The multilevel cache system of claim 10, wherein a tag stored in the third array identifies the contents of a set in the first array and second array.

12. The multilevel cache system of claim 10, wherein the tag array is configured to contain a plurality of entries, and wherein each entry in the tag array contains a presence bit to indicate whether a corresponding set in the first array contains the same information as a corresponding set in the second array.

13. The multilevel cache system of claim 12, wherein the second array contains a plurality of ways, and wherein each entry in the tag array contains a presence-way bit to indicate which way in the second array, if any, contains information that is present in a corresponding set in the first array.

14. A computer system, comprising:

a central processing unit;

a merged tag array coupled to the central processing unit;

a first data array coupled to the central processing unit; and

a second data array coupled to the merged tag array.

15. The computer system of claim 14, further comprising a processor state control component coupled to the central processing unit and to the merged tag array.

16. The computer system of claim 15, wherein the merged tag array is configured to contain directory information for the first data array and second data array.

17. The computer system of claim 16, wherein:

the first data array contains a plurality of sets and the second data array contains a plurality of sets;

the merged tag array contains a plurality of entries, each of which corresponds to a set in the first data array and to one or more sets in the second data array; and

each entry in the merged tag array contains a presence field indicating whether a corresponding set in the second data array contains a copy of information that is also present in a corresponding set in the first data array.

18. The computer system of claim 17, wherein:

the second data array further contains a plurality of ways; and

the merged tag array further contains a presence-way field that indicates which way in the second data array contains a copy of information present in a corresponding set in the first data array.

19. A method of retrieving information from a multilevel cache system, comprising:

issuing a request for information to a first data array, a second data array, and a merged tag array at substantially the same time;

receiving information stored in a first data array location corresponding to the request;

tentatively processing an instruction that consumes the information received;

determining from the merged tag array whether the request was a cache hit in the first data array and whether the request was a cache hit in the second data array; and

retiring the instruction tentatively processed if the request was a cache hit in the first data array.

20. The method of claim 19, further comprising when it was determined that the request was a cache miss in the first data array:

flushing the instruction tentatively processed;

loading the information from the second data array into the first data array if the request was a cache hit for the second data array;

forwarding the request to another level of memory hierarchy if the request was a cache miss for the second data array; and

replaying the instruction tentatively processed.

21. The method of claim 19, wherein determining if the request generated a cache miss for the first data array comprises:

checking a presence field in a entry of the merged tag array corresponding to the request to determine if a corresponding set of the first data array contains a copy of information present in one of the corresponding sets of the second data array; and

determining if a copy of the requested information is contained in an set of the second data array that both corresponds to the request and contains a copy of the information present in a corresponding set of the first data array.

22. The method of claim 19, wherein:

the second data array contains a plurality of ways; and

determining if an set in the second data array contains a copy of the information present in a corresponding set of the first data array comprises determining if a tag identified by a presence-way field matches a portion of the memory address requested.

23. The method of claim 19, further comprising:

sending the request for information to a single level translation lookaside buffer at substantially the same time as it is sent to the first data array, second data array, and merged tag array, wherein the single level translation lookaside buffer contains all available memory address translations;

checking the single level transaction lookaside buffer to determine whether the request is authorized; and

transferring control to an exception handler if the request is not authorized.

24. A method of snooping a multilevel cache system, comprising:

sending a snoop request to a merged tag array;

checking the merged tag array to determine if a copy of information corresponding to the request is present in an entry of the multilevel cache; and

if a copy of the information is present in an entry of the multilevel cache, modify the entry of the merged tag array corresponding to the request to invalidate the entry.

25. The method of claim 24, wherein:

the multilevel cache contains at least a first data array and a second data array;

the second data array contains a plurality of ways, and each entry of the merged tag array contains a valid field for each way corresponding to the entry; and

modifying the entry of the merged tag array corresponding to the request to invalidate the entry comprises invalidating a presence field in the entry and invalidating the valid field in the entry for the way that corresponds to the request.